MMA Technical Standards Board/ AMEI MIDI Committee

Letter of Agreement for Recommend Practice

Mobile Phone Control Message (RP-046)

Universal Real Time System Exclusive

Source: MMA Mobile Working Group, Beatnik

Abstract:

Defines new Universal Real Time System Exclusive message with extensible framework for controlling all non-musical capabilities in mobile phone-oriented player devices, including vibrators and LEDs.

Background:

MMA / AMEI specs are currently very limited in their ability to use MIDI to control non-music output devices commonly used in mobile devices such as mobile phones: the SP-MIDI 5-24 Voice Profile for 3GPP (RP-035) includes a way to control one vibrator, and there are no other examples. In the mobile industry, there have been multiple proprietary methods of using MIDI to control not only vibrators, but also LEDs and other non-musical output devices. This kind of functionality is considered desirable for multiple types of mobile content, especially mobile phone ring tones. Rather than incorporate a proprietary solution into an MMA/AMEI specification, the Mobile Working Group proposed this new Universal System Exclusive mechanism.

Publication Plan:

Since this is a Universal System Exclusive message that could be used in many different contexts (including SMF, XMF, or live MIDI message stream), it should be published on www.midi.org with other MIDI 1.0 Specification enhancements, and should be incorporated into any future update of the MIDI 1.0 Specification document.

Details:

1. Introduction

This proposal defines a new Universal Real Time System Exclusive message as a standardized general framework for controlling anything beyond MIDI notes in a mobile phone-oriented player device. Each SysEx message consists of one command, addressed to one destination device such as one particular LED or one particular vibrator.

1.1 Device Classes

A set of standard destination Device Classes is defined: Vibrators, LEDs, Displays, and Keypads. This list can be extended by MMA/AMEI in the future.

Within a given Device Class, instances of the class are addressed by a zero-based index, where index 0 represents the primary instance. For example, Device Class 2 is Vibrators, and vibrator number 0 is the primary vibrator (the telephone ring vibrator), so to address the ring vibrator, use class 2 and index 0. On a mobile phone with 2 vibrators, vibrator index 1 would also be available. Device Class 127 acts as a 'call all' that matches all possible device classes. Similarly, device index 127 acts as a 'call all' that matches all possible instances of the given device class.

© 2006 MIDI Manufacturers Association Incorporated • www.midi.org

The set of device classes is also manufacturer-extensible via the Manufacturer-Specific Device Class (1) which uses the manufacturer's MIDI Manufacturer ID plus a further manufacturer-specific DestClass byte. Messages that target unrecognized Manufacturer-Specific Device Classes should be expected, and properly ignored.

1.2 Commands

A set of standard Commands is also defined: Reset, On, Off, Follow MIDI Channels, Set Color RGB, and Set Level. This list can be extended by MMA/AMEI in the future. The action of every command is defined for every standard Device Class.

The command set is also manufacturer-extensible via the Manufacturer Specific Command (CmdID 1) which uses the MIDI Manufacturer ID and may also use further manufacturer-specific data bytes if the manufacturer so deems. Messages that use unrecognized Manufacturer Specific Commands should be expected, and properly ignored.

1.3 Content Portability

While this specification defines the control of vibrators, LEDs, displays and keypads, it is important to note that a particular platform is not required to have all these device types (device classes), nor is a minimum number of devices (device indices) for each class explicitly set. Thus, if content authors wish to attain more portability of their content between different devices, it is recommended to minimize the number of devices controlled, and to use device indices starting from the primary instance (i.e. deviceIndex 0). Even if particular device indices are reserved (e.g. individually-controlled LEDs under the keypad), the author cannot assume that all platforms will contain, or be able to control, these device indices. If the content calls for control of a device that does not exist on a particular platform, the instructions for that device shall be ignored by the implementation.

2. Technical Definitions

2.1. Message Format

The SysEx <sub-ID#1> field for this message is 0Ch, and the value of <sub-ID#2> is 00h.

```
F0 7F <phone ID> 0C 00                    // Universal SysEx message header
<deviceClassID> <deviceIndex> <cmdID> <dataBytes>
F7                                        // End of SysEx
```

DeviceClassIDs and deviceIndexes are described in section 2.2.

DeviceIndexes are described in section 2.3.

CmdIDs and the format of any dataBytes are described in section 2.4.

Example messages are shown in section 3.

2.2. DeviceClassID Definitions

deviceClass ID	Meaning
0	**Reserved** for future MMA / AMEI Definition
1	**Manufacturer-Specific DeviceClasses** Followed by 1-byte or 3-byte MIDI Manufacturer ID, then mfr-specific 1-byte DeviceClass
2	**Vibrator** deviceIndex #0 is the telephone ring vibrator

© 2006 MIDI Manufacturers Association Incorporated • www.midi.org

3	**LED** deviceIndex #0 is the primary LED If a numeric keypad with a separate, individually addressable LED for each number key (and optionally the # and * keys) is present: deviceIndex 100 – 108: Keys 1 through 9 deviceIndex 109: '*" deviceIndex 110: '0" deviceIndex 111: '#"
4	**Display** deviceIndex #0 is the primary display
5	**Keypad** deviceIndex #0 is the primary keypad Note: If a handset has multiple, individually addressable lights under a keypad, they should be made available as multiple LED deviceIndexes, not Keypad deviceIndexes.
6-126	**Reserved** for future MMA / AMEI Definition
127	**All DeviceClasses** present ('call all')

2.3. DeviceIndex Definitions

deviceIndex	Meaning
0-126	**Valid common deviceIndex** For all DeviceClasses, deviceIndex 0 is always the primary instance
127	**All DeviceIndexes** present for the given DeviceClass ('call all')

2.4. CmdID Definitions

cmdID	Meaning and Data Bytes
0	**Reserved** for future MMA / AMEI Definition
1	**Manufacturer-Specific Command** Followed by 1-byte or 3-byte MIDI Manufacturer ID, then variable number of mfr-specific cmd and/or data bytes
2	**Reset** No data bytes For Vibrators: Turn vibrator off (& set default frequency if available) For LEDs: Turn LED off (& set default color if available) For Displays & Keypads / Keyboards: Reset to default
3	**On** No data bytes For Vibrators: Turn vibrator on For LEDs: Turn LED on For Displays & Keypads / Keyboards, this affects BG/border light
4	**Off** No data bytes For Vibrators: Turn vibrator off For LEDs: Turn LED off For Displays & Keypads / Keyboards, this affects BG/border light
5	**Follow MIDI Channels** Variable number of data bytes: List of, Channel, LowNote, HighNote triples (1 byte for each number), or empty list to cancel. MIDI channel numbers must be 0-15, and note numbers must be 0-127. To change MIDI response, send a

© 2006 MIDI Manufacturers Association Incorporated • www.midi.org

	new list. Every MIDI Note On message in the indicated MIDI Channel with a note number in the range between LowNote and HighNote (inclusive) causes the same action as the On command (CmdID 3). Every MIDI Note Off message (or Note On message with velocity of 0) in the indicated MIDI Channel with a note number in the range between LowNote and HighNote (inclusive) causes the same action as the Off command (CmdID 4). For Vibrators: Turn vibrator on and off For LEDs: Turn LED on and off For Displays & Keypads / Keyboards, this affects BG/border light
6	**Set Color RGB** 3 data bytes: R, G, B [7 bits each] (Black = 00,00,00 White = 7F,7F,7F) For Vibrators: Ignored For LEDs: Sets LED color (if available) For Displays & Keypads / Keyboards, sets BG/border color (if available)
7	**Set Level** 1 data byte: Level [7 bits] (max = 7F, min/off = 00) For Vibrator: Sets level (if available) For LEDs: Sets LED brightness (if available) For Displays & Keypads / Keyboards: Sets BG/border light brightness (if available)
8-127	**Reserved** for future MMA / AMEI Definition

3. Example Messages

3.1. Reset All Available Devices in the Mobile Phone

```
F0 7F <phone ID> 0C 00      // Universal Sys Ex header
7F 7F                       // All device indexes for all device classes
02                          // Reset cmd
F7                          // End of SysEx
```

3.2. Turn Phone Ring Vibrator On

```
F0 7F <phone ID> 0C 00      // Universal Sys Ex header
02 00                       // Vibrator #0
03                          // On cmd
F7                          // End of SysEx
```

3.3. Turn Phone Ring Vibrator Off

```
F0 7F <phone ID> 0C 00      // Universal Sys Ex header
02 00                       // Vibrator #0
04                          // Off cmd
F7                          // End of SysEx
```


© 2006 MIDI Manufacturers Association Incorporated • www.midi.org

3.4. Set LED #4 Color to Purple

```
F0 7F <phone ID> 0C 00      // Universal Sys Ex header
03 04                       // LED #4
06 7F 00 7F                 // Set Color RGB cmd, full R, zero G, full B
F7                          // End of SysEx
```

3.5. Make LED #4 Follow MIDI Channels 4 (All Notes) & 8 (Middle C Only)

```
F0 7F <phone ID> 0C 00      // Universal Sys Ex header
03 04                       // LED #4
05                          // Follow MIDI Channels cmd
03 00 7F                    // MIDI ch 4, lowNote=0, highNote=127
07 40 40                    // MIDI ch 8, lowNote = 64, highNote = 64
F7                          // End of SysEx
```

4. Player Behavior Requirements

4.1. Vibrator and LED Control

4.1.1 On/Off Reference Counting During Playback

On/off switching of each vibrator and each LEDs should observe a 'reference count' model, to ensure that once turned on, the device stays in the ‘on’ state until the number of 'off' events encountered equals the number of 'on' events previously encountered. Any On command, or MIDI Note On message in the case of the Follow MIDI Channel command, increments the reference count. Any Off command, or MIDI Note Off message (or Note On with velocity of 0) decrements the reference count. The minimum value of the reference count is zero, in other words the reference count can’t be decremented to a value lower than zero. The maximum value of the reference counter must be at least 255. The reference counter must not be incremented past the maximum value. The reference counter must be set to 0 at the beginning of playback.

4.1.2. Save Initial State and Restore After Playback

At the end of playback for any reason, the player should restore every vibrator and LED to the same on/off status it had before playback started. If any of the LEDs allows control of color, the original color must be restored at the end of playback for any reason.

4.2. Message Execution Order in SMFs

While the Standard MIDI File (SMF) specification [1] does not mandate the order in which MIDI messages that occur at the same tick should be processed, predictable order of execution for Mobile Phone Control messages is important. Therefore players supporting Mobile Phone Control messages MUST process MIDI events within each SMF track in the order in which they appear in the SMF, even when the delta time between messages is zero.

© 2006 MIDI Manufacturers Association Incorporated • www.midi.org

To guarantee execution order, content creators should place all Mobile Phone Control SysEx messages in the same SMF track, and/or separate all Mobile Phone Control SysEx messages in time by at least one SMF tick.

4.3. Exclusive Ownership of Controlled Devices

A phone should not play more than one piece of content using Mobile Phone Control System Exclusive messages at the same time, since the addressing mechanism assumes that each piece of content owns all the devices on the phone. This is an output device policy.

4.4. Minimum On and Off Times

For LED devices, the player may enforce a minimum on time before turning off, and/or a minimum off time between on states. This is to compensate for potentially careless MIDI authoring practices when the Follow MIDI Channels command is used. In particular, it is common for drum sequences to have Note Off messages too soon after the corresponding Note On messages, or for the Note Off messages to be missing entirely.

5. References

[1] *"Standard MIDI Files"*, RP-001,in The Complete MIDI 1.0 Detailed Specification, Document Version 96.1, MIDI Manufacturers Association, Los Angeles, CA, USA, 1996

© 2006 MIDI Manufacturers Association Incorporated • www.midi.org